

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF THE CLAIMS:

1. (Canceled)
2. (Currently Amended) The compiler apparatus according to claim 1 5, wherein said exception handler throw ~~section~~ means throws a branch instruction for causing a shift to said common processing in said clone exception handler and causes a shift to said common processing with said branch instruction thrown.
3. (Currently Amended) The compiler apparatus according to claim 1 5, wherein said exception selection ~~section~~ means selects a set of exceptions whose frequency of throw in said multiple-catching exception handler is more than a predetermined reference frequency and which is shifted to said common processing.
4. (Original) The compiler apparatus according to claim 3, wherein, as a frequency with which exceptions are thrown in said multiple-catching exception handler, said exception selection ~~section~~ means detects the number of times that any of said set of exceptions is thrown in said multiple-catching exception handler per the number of execution of said multiple-catching exception handler.
5. (Currently Amended) A The compiler apparatus for improving maintainability and robustness of programs by optimizing exception handling in a target program according to claim 1, wherein
an exception handler detection means for detecting, from exception handlers that catch exceptions thrown in said target program, a multiple-catching exception handler that catches a plurality of different exceptions and rethrow the caught exceptions;

an exception selection means for selecting a set of exceptions that are to be shifted to common processing through rethrow of the exception by said multiple-catching exception handler from among said plurality of exceptions caught by said multiple-catching exception handler detected;

an exception handler throw means for throwing a clone exception handler that catches the set of exceptions selected by said exception selection means instead of said multiple-catching exception handler and shifting it to said common processing;

said common processing includes catching exceptions that are thrown while processing a function that has been called with a function call outside said function call;

said multiple-catching exception handler catches exceptions inside said function call; and

said exception selection ~~section~~ means selects said set of exceptions further on condition that depth of nesting of function call from said common processing down to said multiple-catching exception handler is more than a predetermined number.

6. (Currently Amended) A ~~The compiler apparatus according to claim 1, for improving~~
maintainability and robustness of programs by optimizing exception handling in a target program comprising:

an exception handler detection means for detecting, from exception handlers that catch exceptions thrown in said target program, a multiple-catching exception handler that catches a plurality of different exceptions and rethrow the caught exceptions;

an exception selection means for selecting a set of exceptions that are to be shifted to common processing through rethrow of the exception by said multiple-catching exception handler from among said plurality of exceptions caught by said multiple-catching exception handler detected;

an exception handler throw means for throwing a clone exception handler that catches the set of exceptions selected by said exception selection means instead of said multiple-catching exception handler and shifting it to said common processing.

wherein the exception selection ~~section~~ means selects the set of exceptions further on condition that the number of other exception handlers through which processing shifts from the multiple-catching exception handler to the common processing is more than a predetermined number.

7. (Currently Amended) A ~~The compiler apparatus according to claim 1, wherein~~ for improving maintainability and robustness of programs by optimizing exception handling in a target program comprising:

an exception handler detection means for detecting, from exception handlers that catch exceptions thrown in said target program, a multiple-catching exception handler that catches a plurality of different exceptions and rethrow the caught exceptions;

an exception selection means for selecting a set of exceptions that are to be shifted to common processing through rethrow of the exception by said multiple-catching exception handler from among said plurality of exceptions caught by said multiple-catching exception handler detected;

an exception handler throw means for throwing a clone exception handler that catches the set of exceptions selected by said exception selection means instead of said multiple-catching exception handler and shifting it to said common processing;

said common processing includes catching exceptions that are thrown while processing a function that has been called with a function call outside said function call;

said multiple-catching exception handler catches exceptions inside said function call; and

said exception selection ~~section~~ means selects said set of exceptions further based on a depth of nesting function call from said common processing down to said multiple catching exception

handler and the number of other exception handlers through which processing shifts from said multiple-catching exception handler to said common processing.

8. (Currently Amended) The compiler apparatus according to claim 4, wherein

said exception handler detection ~~section~~ means detects two said multiple-catching exception handlers: one said multiple-catching exception handler and another said multiple-catching exception handler for catching at least one exception thrown in said one multiple-catching exception handler;

said exception selection ~~section~~ means selects a set of exceptions to be shifted to said common processing by rethrowing an exception caught in said another multiple-catching exception handler from among a plurality of exceptions caught by said one multiple-catching exception handler; and

said exception handler throw ~~section~~ means throws each of two said clone exception handlers that correspond to each of said two multiple-catching exception handlers and causes each of said corresponding two clone exception handlers to catch the set of exceptions selected by said exception selection ~~section~~ means instead of each of said two multiple-catching exception handlers.

9. (Canceled)

10. (Currently Amended) ~~The compiler program according to claim 9, wherein~~ A computer program product comprising a computer usable medium having computer readable program code stored therein for causing a computer to execute method steps for compiling a target program, said method steps comprising:

detecting, from exception handlers that catch exceptions thrown in said target program, a multiple-catching exception handler that catches a plurality of different exceptions and rethrow the caught exceptions;

selecting a set of exceptions that are to be shifted to common processing through rethrow of the exception by the multiple-catching exception handler from among said plurality of exceptions caught by said multiple-catching exception handler detected;

throwing a clone exception handler that catches the set of exceptions selected by said selecting instead of said multiple-catching exception handler and shifting it to said common processing;

said common processing includes catching exceptions that are thrown while processing a function that has been called with a function call outside said function call; and

said multiple-catching exception handler catches exceptions inside said function call; and

~~said exception-selection-section-~~ selecting selects said set of exceptions further based on a depth of nesting function call from said common processing down to said multiple catching exception handler and the number of other exception handlers through which processing shifts from said multiple-catching exception handler to said common processing.

11. (Currently Amended) The ~~compiler program~~ computer program product according to claim 9 10, wherein

~~said exception handler-detection-section-~~ detecting detects two said multiple-catching exception handlers: one said multiple-catching exception handler and another said multiple-catching exception handler for catching at least one exception thrown in said one multiple-catching exception handler;

~~said exception-selection-section~~ selecting selects a set of exceptions to be shifted to said common processing by rethrowing an exception caught in said another multiple-catching exception handler from among a plurality of exceptions caught by said one multiple-catching exception handler; and

said ~~exception handler throw section~~ throwing throws each of two said clone exception handlers that correspond to each of said two multiple-catching exception handlers and causes each of said corresponding two clone exception handlers to catch the set of exceptions selected by said ~~exception selection section~~ selecting instead of each of said two multiple-catching exception handlers.

12. (Canceled)

13. (Currently Amended) A compiling method for causing a computer to operate as a compiler apparatus that optimizes exception handling in a target program as a program to be compiled, comprising:

detecting, from exception handlers that catch exceptions thrown in said target program[,], a multiple-catching exception handler catching a plurality of different exceptions and rethrowing the caught exceptions;

selecting a set of exceptions that are to be shifted to common processing through rethrow of the exceptions from among said plurality of exceptions caught;

throwing a clone exception handler catching the set of exceptions selected by said step of selecting; and

shifting the set of exceptions to said common processing;

said common processing includes catching exceptions that are thrown while processing a function that has been called with a function call outside said function call;

said multiple-catching exception handler catches exceptions inside said function call; and

said selecting selects said set of exceptions further on condition that depth of nesting of function call from said common processing down to said multiple-catching exception handler is more than a predetermined number.

14. (Canceled)

15. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for optimizing exception handling in a target program as a program to be compiled, said method steps comprising the steps of claim-13:

detecting, from exception handlers that catch exceptions thrown in said target program, a multiple-catching exception handler catching a plurality of different exceptions and rethrowing the caught exceptions;

selecting a set of exceptions that are to be shifted to common processing through rethrow of the exceptions from among said plurality of exceptions caught;

throwing a clone exception handler catching the set of exceptions selected by said step of selecting;

shifting the set of exceptions to said common processing;

said common processing includes catching exceptions that are thrown while processing a function that has been called with a function call outside said function call;

said multiple-catching exception handler catches exceptions inside said function call; and

said selecting selects said set of exceptions further on condition that depth of nesting of function call from said common processing down to said multiple-catching exception handler is more than a predetermined number.

16. (Canceled)

17. (New) A compiling method for causing a computer to operate as a compiler apparatus that optimizes exception handling in a target program as a program to be compiled, comprising:

detecting, from exception handlers that catch exceptions thrown in said target program, a multiple-catching exception handler catching a plurality of different exceptions and rethrowing the caught exceptions;

selecting a set of exceptions that are to be shifted to common processing through rethrow of the exceptions from among said plurality of exceptions caught;

throwing a clone exception handler catching the set of exceptions selected by said step of selecting;

shifting the set of exceptions to said common processing,

wherein said selecting selects the set of exceptions further on condition that the number of other exception handlers through which processing shifts from the multiple-catching exception handler to the common processing is more than a predetermined number.

18. (New) A compiling method for causing a computer to operate as a compiler apparatus that optimizes exception handling in a target program as a program to be compiled, comprising:

detecting, from exception handlers that catch exceptions thrown in said target program, a multiple-catching exception handler catching a plurality of different exceptions and rethrowing the caught exceptions;

selecting a set of exceptions that are to be shifted to common processing through rethrow of the exceptions from among said plurality of exceptions caught;

throwing a clone exception handler catching the set of exceptions selected by said step of selecting;

shifting the set of exceptions to said common processing;

said common processing includes catching exceptions that are thrown while processing a function that has been called with a function call outside said function call;

said multiple-catching exception handler catches exceptions inside said function call; and

said selecting selects said set of exceptions further based on a depth of nesting function call from said common processing down to said multiple catching exception handler and the number of other exception handlers through which processing shifts from said multiple-catching exception handler to said common processing.

19. (New) The compiler apparatus according to claim 5, wherein said compiler apparatus includes a function of optimizing said target program to be compiled before starting to execute said target program as well as a function of optimizing said target program during execution of said target program.